7.3 Schutzebenen und Barrieren

Wie bereits vorher beschrieben, können Fehler und funktionale Effekte in beliebiger Form zu Störfällen und Unfällen führen. Die Effekte und Fehler können in horizontaler Ebene (zum Beispiel vom Sensor über die Verarbeitung hin zum Aktuator) propagieren, oder vertikal vom physikalischen Effekt zum Beispiel in der Elektronikhardware hin zu einem Störfall oder Unfall. Daher ist es notwendig, Schutzebenen oder Barrieren einzuplanen.

7.3.1 Fehler- und Risiko-Pyramide

Bei automatisierten Fahrfunktionen, die man in dem Kontext der jeweiligen Verkehrssituation betrachten muss, wird schnell transparent, dass eine Fehlerpyramide sehr viele Ebenen hat.

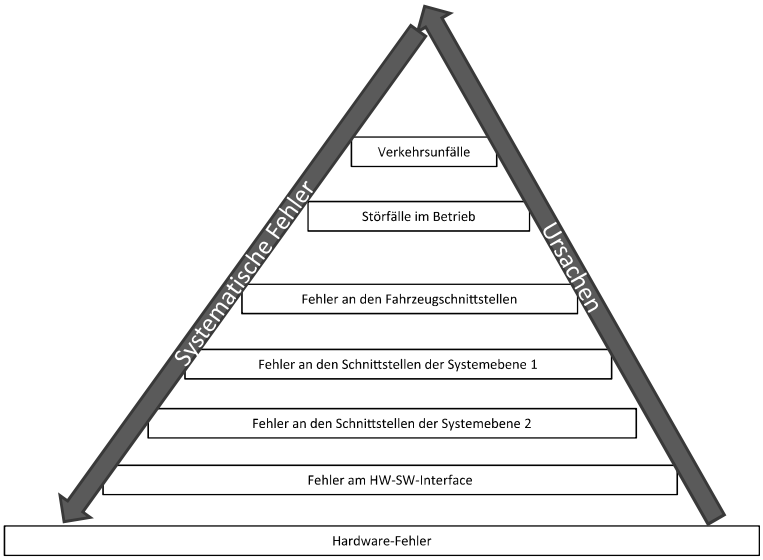

Bild 7.28 Fehler- und Risikopyramide

Die systematischen Fehler (nicht nur die gemäß ISO 26262), wie Spezifikationsfehler, falsche Einschätzungen von Situationen, Irrtümer, nicht geeignete Komponenten oder Algorithmen und Abhängigkeiten, vererben sich bis in die Hardware-Design-Entscheidungen beziehungsweise die physikalische Ebene hinunter. Defekte und die Auswirkungen von Designfehlern gehen von der physikalischen Ebene

über alle Fehler an den Schnittstellen nach oben, bis es zu Unfällen im Verkehr kommt. Eine solche Pyramide kann nicht wirklich von Menschen, auch mit allen technischen Hilfsmitteln, über Ursache-Wirkungs-Analysen analysiert werden.

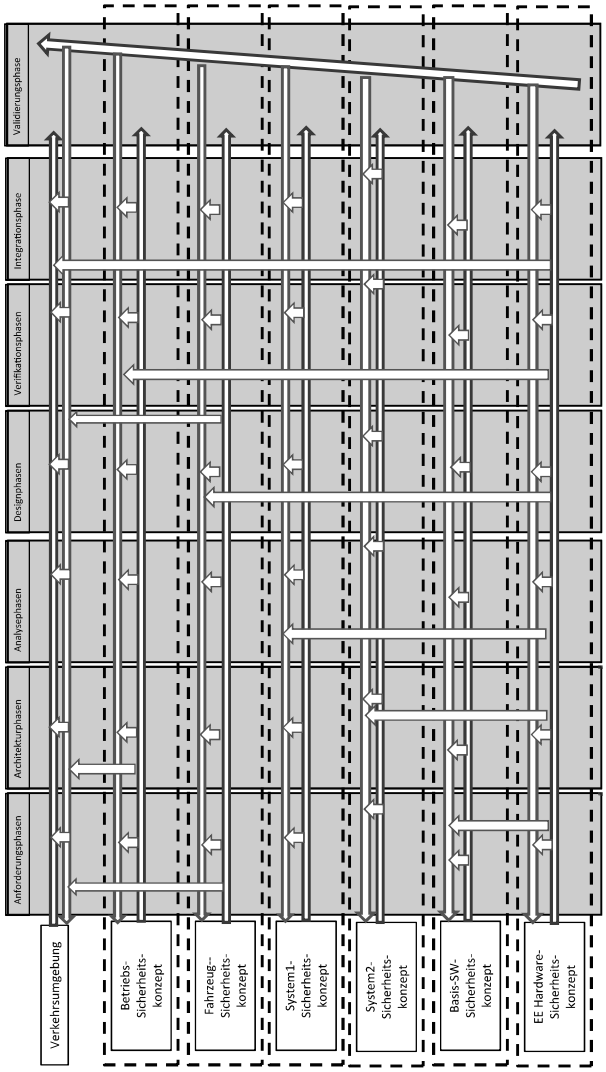

Bild 7.29 Ebenen der horizontalen Systemabsicherung

Grundsätzlich muss von den Maßnahmen in den verschiedenen Ebenen, ausgehend von Maßnahmen in der Verkehrsumgebung bis zu Maßnahmen auf der unteren physikalischen Ebene (z. B. EE-Hardware), derselbe grundsätzliche Prozess abgearbeitet werden.

Der Prozess muss die üblichen Phasen betrachten:

- Anforderungsphase,
- Architekturphase,
- Analysephase,
- Designphase,
- Verifikationsphase,
- Integrationsphase,
- Validationsphase.

Die Phasen werden wie in jeder Entwicklung keiner eindeutigen Sequenz folgen, die Aktivitäten müssen aber in jeder dieser Ebenen durchgeführt werden.

Sämtliche Architektur- oder Designentscheidungen, logischen Zuordnungen, organisatorischen Schnittstellen sind rein willkürlich und entstammen den Rahmenbedingungen und Einschränkungen des Produktentstehungsprozesses.

- Funktionales Verhalten,
- Systemverhalten,
- zeitliche Abhängigkeiten,
- eingeschränkte Fähigkeiten von System, Komponenten und Algorithmen sowie
- Fehler, Fehlverhalten und Fehlerpropagationen

werden durch solche in der Entwicklung festgelegten Schnittstellen nicht beeinflusst.

Einen solchen Zusammenhang hat Reason 1990 bereits mit seinem Schweizer-Käse-Modell beschrieben.

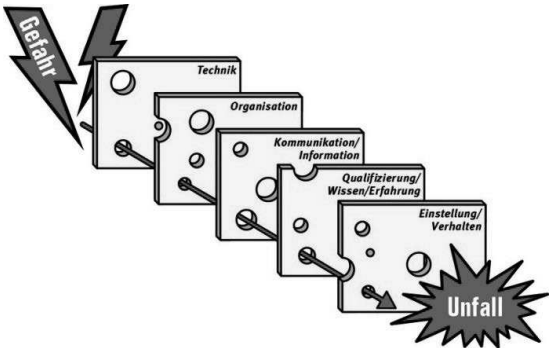

Bild 7.30 Schweizer-Käse-Modell (Quelle: Reason 1990)

Die Kette zeigt Fehler an den Organisationsschnittstellen auf. Die Kette hier verdeutlicht, dass latente Fehler (Fehler, die im Produktdesign schlummern) über

- die Technik zur
- Organisation, weiter zu
- Fehlern in der Überwachung, zu
- Fehlverhalten des Systems oder Flugzeugs oder Fahrzeugs bis hin zu
- Unfällen führen.

Ähnliche Ketten sehen die Ursache

- in der Technik,
- der Organisation,
- der Kommunikation der Informationen (Spezifikation, Instruktionen etc.),
- der Qualifikation der Agierenden,
- dem Verhalten und der Einstellung der Protagonisten.

Protagonisten können Hersteller, Vertreiber, Händler, Anwender und auch Verkehrsteilnehmer als die Gefährdeten sein, aber auch als Ursachen für den Fall, dass durch ihr Verhalten andere gefährdet werden.

7.3.2 Diversität zur Risikoreduzierung

In einigen Veröffentlichungen zur ISO 26262 wurde das Schweizer-Käse-Modell auch für die ASIL-Dekomposition angewendet. Leider wurde in keiner der Veröffentlichungen darauf hingewiesen, dass es das Ergebnis einer Analyse und eine Architekturmaßnahme ist, die Käsescheiben so anzuordnen, dass zum Beispiel Fehler nicht durch alle Löcher durchpropagieren. Weiter hat man in anderen Branchen schon festgestellt, dass man nicht einfach Käsescheiben aus der französischen Schweiz und der italienischen Schweiz als diversitäre Käsesorten deklarieren und davon ausgehen kann, dass sich dadurch die Löcher an verschiedenen Stellen befinden. Dies gilt natürlich für jeden anderen löchrigen Käse genauso. Hier kommt man wieder schnell zu dem Schluss, dass ein diversitäres Systemdesign oder diversitäre Funktionen und Algorithmen überhaupt keinen generischen Vorteil für die Sicherheitstechnik haben. Homogene Redundanzen lassen sich sogar viel schneller synchronisieren und damit vergleichbar machen, damit man potentielle Quellen von latenten Fehlern vor der Fehlerauswirkung entdecken kann. Dies heißt nicht, dass man Objekterkennungen oder auch Zustandserkennungen auf verschiedenen logischen Prinzipien erstellen sollte.

Mikrocontroller und Mikroprozessoren arbeiten weitgehend auf demselben Prinzip, daher kann selbst die parallele Verwendung von Mikrocontrollern und Mikroprozessoren zu sehr vielen ähnlichen Verhaltensweisen führen, die eine Erkennung von systematischen Fehlern nicht erlauben werden. Eine parallele Verwendung von

Mikrocontrollern verschiedener Hersteller wird erst recht kein Ausschlusskriterium für systematische Fehler sein, weil nur Mutmaßungen existieren können, wo sich die Lücken im Design oder Ähnliches durch die redundante Verwendung finden.

Die Druckkesselverordnung (zum Beispiel Druckgeräterichtlinie 2014/68/EU) geht auf die Bierbrauer von Mannheim und Heilbronn zurück. Hier legte man vor mehr als 100 Jahren fest, dass die Temperatur, der Druck und der Flüssigkeitsstand zu messen sei. Die Abhängigkeit dieser Größen ist durch das Design des Druckkessels gegeben. Wird eine dieser Größen außerhalb der Spezifikation für den jeweiligen Betriebsfall betrieben, kann dies zur Gefährdung führen. Der Kessel wird gekühlt, es wird Druck abgelassen, der Füllstand wird reduziert und andere Maßnahmen können eingeleitet werden. Die Bierbrauer haben aber bewusst lieber schlechtes oder weniger Bier gebraut, als die gesamte Brauerei durch eine Explosion zu verlieren. Das gleiche Prinzip gilt für ein Sicherheitssystem für eine Lithium-Ionen-Batterie:

- Temperatur,
- Spannung und
- Ströme

müssen zu jedem Betriebszeitpunkt in einem bestimmten Verhältnis zueinander innerhalb der Spezifikation stehen, sonst riskiert man einen Brand oder die Explosion der Batterie.

Bei dem Braukessel wie bei der Batterie gilt, dass man durch logische physikalische Zusammenhänge immer die Größen gegeneinander überprüfen kann. Somit werden viele systematische und sporadische Fehler entdeckbar. Dieser Effekt wird heute auch bei der Umfelderfassung notwendig sein, da es keine Sensoren gibt, die alle notwendigen Situationen, das Verhalten und so weiter in dem jeweiligen Kontext hinreichend erfassen können.

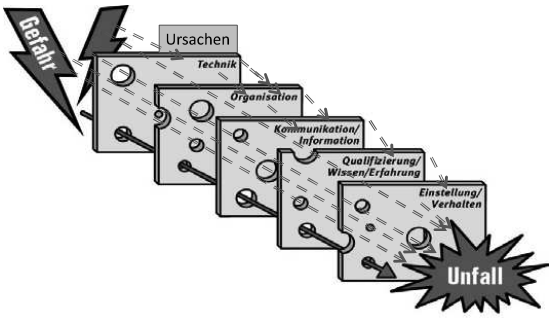

Bild 7.31 Ursache/Wirkung und Propagation von Eigenschaften, Funktionen und Fehlern

Ursachen können durch jede Ebene entstehen und von dieser Ebene aus propagieren. Für welche Propagation von Verhalten und Fehlern tatsächlich kein „Loch im Käse“ vorhanden ist, muss systematisch geplant werden.

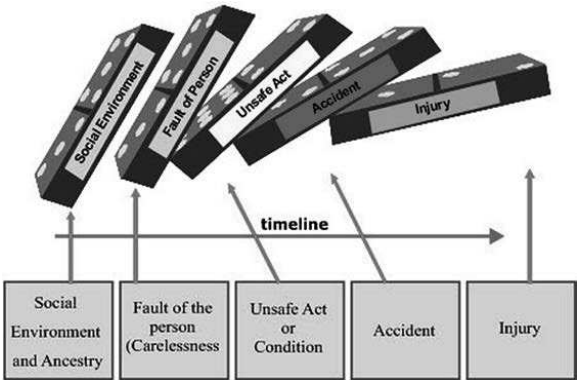

Bild 7.32 Dominoeffekt zu Unfällen und Verletzungen nach Heinrich 1931 (Quelle: Disaster Management Institute, Bhopal, Online Explanation of Domino theory)

Heinrich zeigt die soziale Umgebung und das Umfeld des Entwicklers auf, welches nur einen geringen Einblick in die Anwendungen haben kann; seine Fehler und Irrtümer propagieren aber auch möglicherweise bis hin zu einem Unfall.

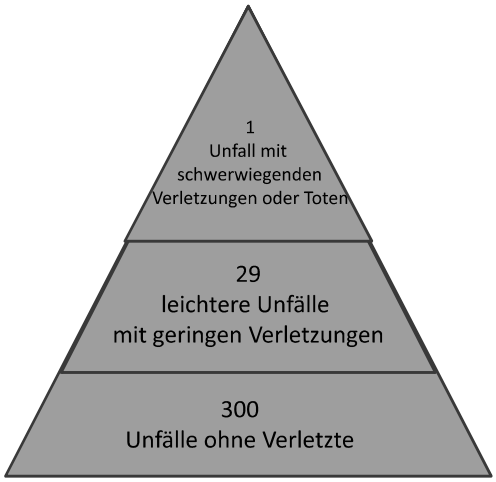

Bild 7.33 Heinrich-Pyramide

Heinrich formulierte auch den Pyramideneffekt, jedoch schloss er darauf, dass man durch effiziente Maßnahmen an den Ursachen viele Fehlereffekte, die zu großen Schäden führen, arbeiten kann. Diese Erkenntnis geht in die FMEA-Methodik ein, bei der man das Vermeiden von Fehlerursachen als effektivste Maßnahme ansieht. Eine der wesentlichen Motivationen der Lebenszyklusansätze war die, dass Ursachen nicht nur in der Technik und im Design verankert sind, sondern auch in vielen anderen Zusammenhängen der Produktentstehung bis hin zum Ausphasen eines Produktes. In heutigen Systemen wird also durch das technische Design kaum ein Produkt definierbar sein, welches frei von Sicherheitsrisiken ist.

Diese Zusammenhänge, die immer wieder aufgezeigt werden, sind heute nicht durch ein Wunder gelöst und werden auch nicht durch so etwas wie künstliche Intelligenz gelöst werden können. Wir fordern heute, dass wir zufällige Hardwarefehler mit einer Diagnosedeckung von 99 % absichern können, und argumentieren immer noch mit Zuverlässigkeitsdaten von Siliziumspeichern aus den 70er Jahren in der Hardwaresicherheit. Wir sprechen von sicheren Hardwarefehlern in Mikroprozessoren, bei denen kein Mensch weiß, welcher Speicher für welche Aufgabe verwendet wird. Geschweige denn, wie diese Fehler in Mikroprozessoren zu einer Gefährdung führen können. Ähnliche Irrtümer wie dieser können an jeder technischen, logischen, organisatorischen Schnittstelle geschehen, ohne dass eine Systematik dies verhindern kann.

Also, wollen wir auf verteilte Entwicklung, künstliche Intelligenz, Hochleistungsrechner verzichten? Im digitalen Zeitalter doch wirklich nicht.

Ein Hardware-Elektroniker kennt seine Hardware am besten, der Software-Ingenieur seine Software, aber die unterschiedlichen Betriebsszenarien und die Reaktionen der Personen im Verkehrsumfeld werden beide nicht systematisch bearbeiten oder analysieren können. Die Effektivität von Maßnahmen in all den Ebenen wird auch nie ein einziger Assessor beurteilen können.

7.3.3 Künstliche Intelligenz und Sicherheit

Wesentlich ist die Frage, was ist künstliche Intelligenz und wozu setzt man diese sinnvoll ein. Schon vor 20 Jahren hat man im Anlagenbau sogenannte „Fuzzy-Regler“ eingesetzt. Diese konnten die Regelstrecke einlernen, aber auch das Verhalten von möglichen Störgrößen erlernen, um damit die Regelparameter zu optimieren. Um einen nach den Regeln der Sicherheitstechnik implementierten Regler abzusichern, ist es vollkommen irrelevant, ob die Regelparameter korrekt oder falsch während der Laufzeit „weglaufen“, die Reglerabsicherung muss eine Abweichung, die zu einer Gefahr werden kann, erkennen und entsprechende Gegenmaßnahmen initiieren. Bei großen kontinuierlichen Anlagen wie einer Destillationskolonne einer Raffinerie wäre es nicht nur ein wirtschaftlicher Schaden, sondern auch ein

Sicherheitsrisiko, eine Anlage plötzlich einfach abzuschalten. Also wird wie bei einem Kernkraftwerk oder einem Flugzeug auf eine sichere Steuerung des Antriebs oder der Anlage umgeschaltet.

Warum dies nicht auch eine Lösung oder gar eine Anforderung für das automatisierte Fahren ist, leuchtet nicht wirklich ein. Einen Regler zum Fahren, Lenken und Bremsen im Fahrzeug würde man nie unüberwacht implementieren, warum will man dies machen, wenn man künstliche Intelligenz einsetzt?

Die heute als schnelle und effiziente Anwendung gepriesene „End2End-KI" wird aus gesellschaftlichen Interessen heraus zu viele Risiken bergen, als dass man eine Akzeptanz dieser Technologie erfahren wird. End2End-KI wird oft als eine Algorithmik verstanden, die von der Erkennung durch Sensoren ausgehend auch die geeignete Reaktion am Aktuator erzeugt.

Dazu muss geklärt werden, ob man KI als Funktion für

- die Nominalfunktion,
- als Schutzfunktion oder
- Sicherheitsmechanismus

einsetzt.

Weiter muss sichergestellt werden, ob

- die Funktion geeignet ist für die Aufgabe,
- die Eigenschaften und Features der KI-Funktion hinreichend sind,
- die KI-Funktionen selbst geeignet sind,
- das System geeignet ist, die Funktionen hinreichend und rechtzeitig auszuführen,
- die Sensoren und Aktuatoren geeignet sind und
- die Regelkreise mit den richtigen Parametern angewendet werden können.

Die weiteren Fragen danach, ob die richtigen Trainingsdaten und Validierungsstrategien angewendet werden können, sind dem nachgelagert.

Die Frage ist: Was ist künstliche Intelligenz (KI)? Bedeutet es: Alles, was der Mensch kann, wird durch eine Maschine umgesetzt? Vielleicht sogar durch eine intelligentere Maschine, als sie der Mensch je sein kann, weil die Summe der Intelligenz aller Menschen in einem System eingebracht wird? Oder ist KI doch nur ein lernender Regler und Speicher, der auf die Reize reagieren kann, die bewusst dem System zugeführt werden? Wie viel Intelligenz bekommt der Mensch durch die Gene von den Vorfahren geliefert und wie viel brauchen wir davon für das automatisierte Fahren?

Künstliche Intelligenz (AI, Artificial Intelligence) ist heute ein Sammelbegriff für

- maschinelles Lernen (Machine Learning),

- Verarbeitung natürlicher Sprache (NLP, Natural Language Processing),
- künstliche neuronale Netze (Deep Learning ist oft Methodik und nicht immer deckungsgleich mit KNN oder anderen Abkürzungen und Bezeichnungen).

Die Begriffe werden heute in unterschiedlichen Branchen und Kontexten verschieden verwendet.

Wissensbasierte Systeme oder sogenannte Expertensysteme bezeichnen modellierte Systeme, die aus formalisiertem Wissen logische Schlüsse ziehen können. Sie werden zum Beispiel in der Medizintechnik zur Diagnostik eingesetzt. Zur Fehleranalyse oder präventiven Diagnose wären sie auch in technischen Systemen sinnvoll einsetzbar.

Musteranalyse und Mustererkennung sind bereits bei der Zutrittskontrolle oder als Iris-Erkennung oder Fingerabdruckerkennung allen Mobiltelefonnutzern bekannt. Für Objekterkennung, Straßenzustandserkennung (AI sollte den aktuellen Reibwert der Straße besser erfassen können als die Reibwertschätzung über Raddrehzahlsensoren bei einem heutigen ABS), Bewegungsprofile von Menschenmengen oder auch Verkehrsströmungen aus der Cloud oder Verkehrsschildererkennung sollte die Technik durchaus sinnvoll sein. Auch bei der Belegung von virtuellen Rasterflächen, wie in der Robotik bekannt, sollten solche Systeme technische Vorteile haben können.

Sprachliche Intelligenz ist zur Sprachsynthese und umgekehrt von Siri und Alexa bereits bekannt, als Schnittstelle zu Navigationssystemen wird dies heute bereits verwendet.

Die Mechanismen der Musteranalyse, Mustererkennung oder Sprecherkennung können auch umgekehrt zur Mustervorhersage genutzt werden. Wie bei einem Kalmanfilter kann durch weitere Messwerte eine Hypothese plausibilisiert oder bewertet werden. Hierzu ist auch der hierarchische Temporalspeicher von Jeff Hawkins bekannt.

In der Robotik manipuliert man auch bewusst künstliche Intelligenz. Minensuche, Schweißroboter und so weiter erkennen anhand von Lernalgorithmen bestimmte Risiken und initiieren entsprechende risikominimierende Maßnahmen.

In der Medizintechnik setzt man auch immer mehr auf lernende künstliche Systeme, um Prothesen für Gliedmaßen optimal an das Verhalten der Menschen anzupassen.

Alexander Wissner-Gross beschreibt ein intelligentes System, welches Entropiekräfte modelliert. Anhand der Umgebung versucht man einen Zustand zur ermitteln und durch eine Aktivität bei möglicher Ausnutzung der verfügbaren Freiheitsgrade einen zukünftigen Zustand zu erreichen. Zur Navigation und zur Ermittlung der optimalen Trajektorie wäre dies eindeutig auch fürs automatisierte Fahren denkbar.

Dies sind nur wenige Beispiele, bei denen künstliche Intelligenz sinnvoll auch für Sicherheitsanwendungen einsetzbar ist oder gar zur Verbesserung der Sicherheit eingesetzt werden kann. Die Ingenieure, die solche Systeme integrieren, sollten die allgemeinen Regeln der Sicherheitstechnik kennen und beherrschen.

Wenn man künstliche Intelligenz einsetzen möchte, dann sollte klar sein, dass der allgemeine breite End2End-Einsatz keine Lösung für die unterschiedlichen Stakeholder der Gesellschaft sein kann. Zielgerichtet muss man prüfen, welche Risiken durch solche Systeme und Funktionen tatsächlich mit welchen Maßnahmen sinnvoll zu beherrschen sind.

Die notwendigen Sicherheitsmechanismen im Fahrzeug, aber auch in der Verkehrslenkung oder Verkehrsüberwachung sollten nach den geeigneten Sicherheitsintegritätsstandards wie ISO 26262 oder IEC 61508 entwickelt werden.

7.3.4 Mehrebenenabsicherung

Aber wie sind wir dann vor bereits 50 Jahren auf den Mond gekommen? Ist es wirklich nur eine Hollywood-Kulisse und ein Schauspiel, was wir alle gesehen haben? Vermutlich nicht. Was wussten die Protagonisten von damals mehr als wir heute? Wahrscheinlich waren sie demütiger und haben sich nicht von Lobbyismus, Aktienkursen und Marketing-Events blenden lassen.

Eine der ältesten Erkenntnisse des Systemengineerings ist, dass man unerwünschte Effekte und Verhalten vermeidet mit den Maßnahmen, die auch analysierbar und bewertbar sind. Aus dieser Idee entwickelte sich die

- Layer-of-Protection-Analyse (LoPA) oder
- Layer-of-Defense (LoD), auch Line-of-Defense.

In der LoPA geht man davon aus, dass die Ursachen für Risiken in jeder Ebene existieren und alle positiven und negativen Effekte Ursache für Risiken in der höheren Ebene sind. Somit kommt man nicht umhin, in jeder Ebene eine Zielüberprüfung, also eine Validierungsphase in Ergänzung zu den Analysen und Verifikationen, zu ergänzen.

Beginnt man von unten in der physikalischen oder der Hardware-Ebene, bringt die ISO 26262 einige Maßnahmen und Methoden mit, die eine solche Betrachtung unterstützen. Kombiniert man eine System-FMEA und eine quantitative Analyse, so erhält man schon einen Überblick über die möglichen systematischen Fehler und die zufälligen Fehler aus der EE-Hardware. Natürlich müssen aus der EE-Mechanik oder anderen nicht elektrischen Komponenten die Fehler auch betrachtet werden, aber dies ist in allen Qualitätsmanagementsystemen ja sowieso schon so gefordert. Natürlich sind bei Kommunikationssystemen wie CAN-Bus, FlexRay oder Ethernet die entsprechenden Ebenen hinunter bis zur

physikalischen Ebene zu berücksichtigen, jedoch kann man sich hier gut am ISO/OSI-Schichtenmodell orientieren. Handelt es sich um reine Sicherheitsintegritätssysteme, bei denen der energielose Zustand der „sichere Zustand“ ist, können die Metriken der ISO 26262 angewendet werden. Es sollte jedem jedoch klar sein, dass eine PMHF von 10E-8 und einer SPFM von mehr als 99 % nicht heißt, dass die relevanten EE-Fehler nicht auftreten; sie treten nur statistisch gesehen hinreichend selten auf. Davon auszugehen, dass die Fehler nicht über die Systemebene hinaus auch zu Gefährdungen propagieren, wäre ein Irrtum und ein systematischer Fehler.

Geht es jedoch um Risiken wie

- die Gebrauchssicherheit,
- aktive Sicherheitsfunktionen oder Schutzfunktionen,
- die mit der Unverfügbarkeit eines Systems oder einer Funktion einhergehen und so weiter,

wird ein „sicherer Fehler“ womöglich auch ein Fehler sein, der das Potential hat, Ursache für eine Gefährdung zu sein.

Daher wird es Ziel der Validierung sein, abzuwägen, welches Ziel mit der Komponente dem Nutzer oder Anwender versprochen wird. Konsequenterweise benötigt man dann wie bei einer Zertifizierung im Flugzeugbau einen kompletten Safety Case, der das Verhalten im Positiven inklusive aller

- Nominalfunktionen innerhalb ihrer Performancegrenzen,
- aller aktiven Sicherheitsfunktionen und Schutzmechanismen innerhalb der geprüften Betriebsgrenzen,
- aller Verhaltensweisen im Fehlerfall und
- aller Sicherheitsmechanismen und deren geprüften Fähigkeiten, Fehler zu beherrschen, sowie aller
- Trennmechanismen und Barrieren,
- Maßnahmen gegen vorsehbaren und aktiven Missbrauch beinhaltet.

Alles, was vom Hersteller nicht spezifiziert und nachgewiesen ist, bedeutet, dass die Komponente oder das System nicht geeignet ist.

Dieser Prozess wird ebenfalls insbesondere für SW-intensive Systeme an allen darüber liegenden Systemgrenzen notwendig sein, weil schon jemand, der eine Anwendersoftware auf einem System installieren möchte, die Parameter, das Verhalten und die Eigenschaften der Mikrocontroller oder Mikroprozessoren nicht mehr beurteilen kann.

Liefert der Hardwarehersteller keine vollständige Basis-Software, ist es sinnvoll, auch an der Schnittstelle zwischen Basis-Software und Anwender-Software eine solche Trennlinie, also eine Schutzebene, zu installieren.

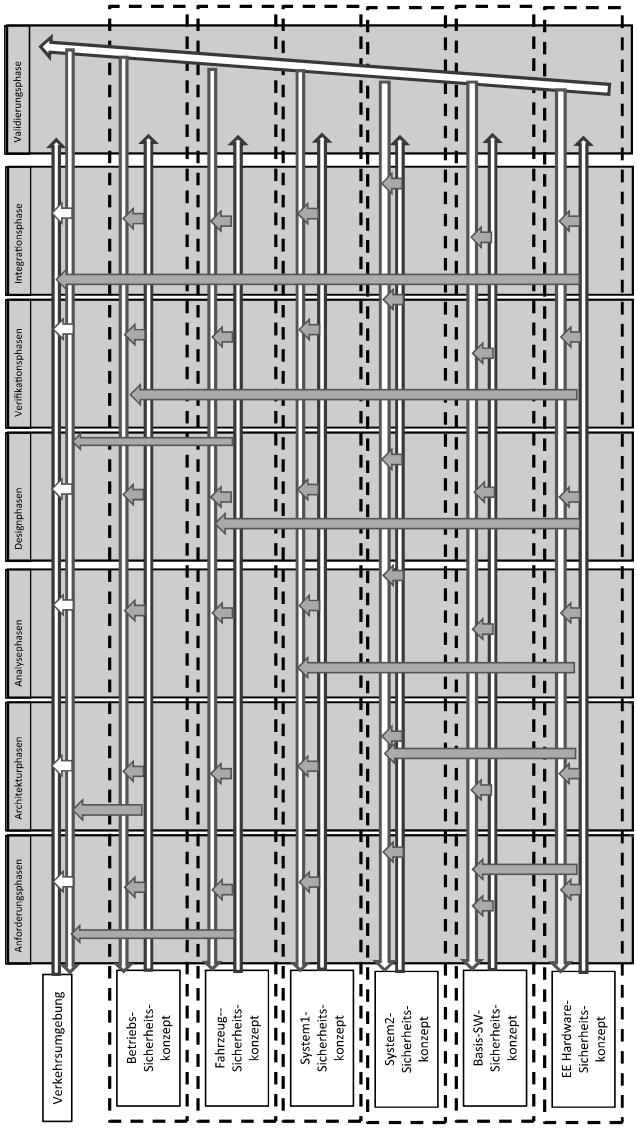

Bild 7.34 Ebenen zur Analyse der Risikoursachen und deren Propagationspotential

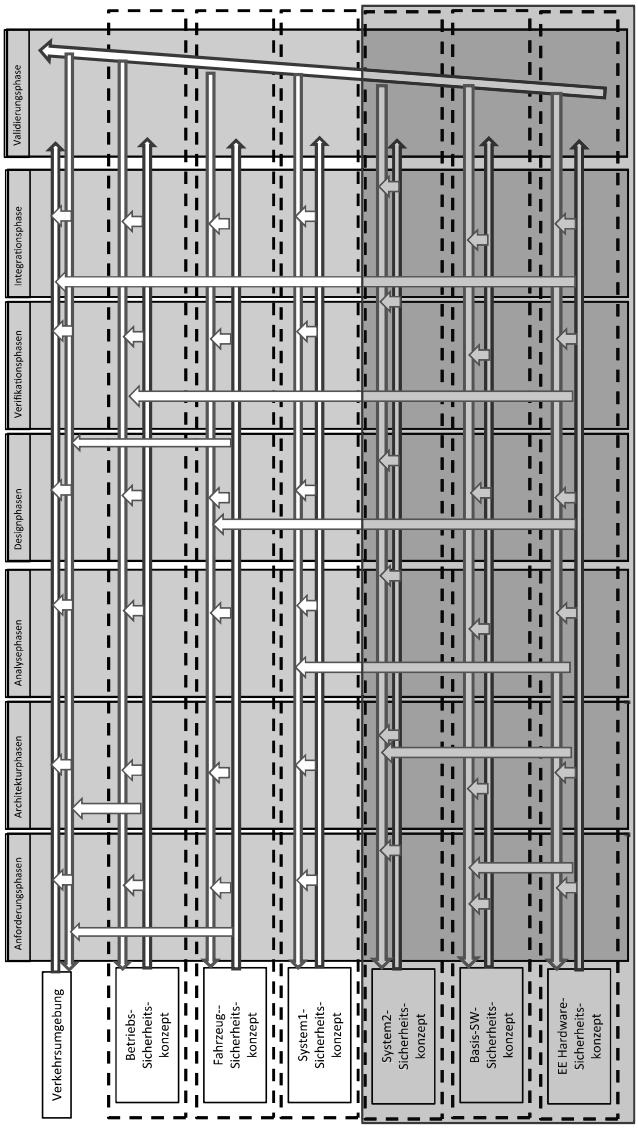

Bild 7.35 Absicherungsebenen in einem Steuergerät

Vergleicht man die Prinzipien, sieht man in einem typischen VDA-3-Ebenen-Sicherheitskonzept, wie EGAS,

- die Ebene 1 für die Nominalfunktion (Motoransteuerung),
- die Ebene 2 für die Funktionsüberwachung (hier vergleichbar mit der System-2-Ebene) und
- die Ebene 3 für die EE-Hardware-Überwachung.

Die Basis-SW ist nicht als Trennmechanismus im Allgemeinen im EGAS ausgeprägt, jedoch wird bei Systemen mit unterschiedlichen ASILs eine solche Ebene notwendig, um Trennebenen oder Barrieren zu implementieren, die die Partitionen zum Beispiel hinreichend trennen.

Die Trennung zwischen Basis-Software und Anwender-Software entsprechend auszubauen, macht schon deswegen Sinn, weil die Entwicklungsparteien hier aus unterschiedlichen Organisationen kommen und somit auch eine Kompetenz dafür angezweifelt werden kann, ob die verantwortliche Organisation für die Anwender-Software überhaupt noch die Verhaltensweisen und Eigenschaften an den Schnittstellen zur Basis-Software und dem Mikrocontroller oder Prozessor beurteilen kann. Einen geeigneten oder „Fähigen Assessor“ zu finden, der so etwas beurteilen kann, mag noch mehr angezweifelt werden.

Ab der Fahrzeugebene wird die Interaktion mit dem Fahrer und anderen Protagonisten, wie Passagieren und anderen Verkehrsteilnehmern, im Vordergrund stehen. Daher wird im Anlagenbau in den oberen Ebenen mehr mit Alarmen und Warnanzeigen etc. gearbeitet. Auf der Betriebsebene werden dann Straßenverkehrsregeln und die Interaktion der Fahrzeuge mit der Infrastruktur im Vordergrund der Maßnahmen stehen, die gewährleisten sollen, dass Fehler der Verkehrsteilnehmer nicht zu Unfällen führen.

7.4 AD-Sicherheitsfunktionen

Um ein Fahrzeug von einem System steuern zu lassen, muss das System die Aufgaben des Fahrers übernehmen. Bei AD-2-Systemen darf der Fahrer praktisch gar nicht die Hände vom Lenkrad lassen und bei AD-Level 3 muss er je nach Funktion innerhalb einer sehr kurzen Zeit die Fahrzeugsteuerung und die Verkehrsraumbeobachtung wieder übernehmen müssen.

Der menschliche Fahrer lernt die unterschiedlichen Situationen und Gegebenheiten während seiner Führerscheinausbildung. In Deutschland machen die Kinder mit zehn Jahren bereits eine Fahrradfahrerausbildung, während der man lernt sich im öffentlichen Verkehrsraum als Fahrradfahrer zu behaupten. Im Kinder-